

PERFORMANCE EVALUATION OF THE DISTRIBUTED IMPLEMENTATION OF A CAR SUSPENSION SYSTEM

Mohamed El Mongi Ben Gaid* Arben Çela*
Saïdou Diallo* Rémy Kocik* Rédha Hamouche*
Abdellatif Reama*

* *COSI Laboratory, ESIEE Paris, FRANCE*

Abstract: In a wide category of embedded systems, sensors, controllers and actuators are located in distant locations. To ensure the control functions, these components need to be interconnected. Usually, communication networks are chosen to support the required communication exchange. In the context of the automotive industry, the Controller Area Network (CAN) bus is widely used because of its interesting properties. In theory, a CAN network can ensure real-time communications, when no transmission errors are detected. However, an increasing number of applications are sharing this bus, generating jitter and delays on the exchanged messages, which can degrade the performance of control applications. In this paper, the distributed implementation of an active suspension controller is modelled in Simulink thanks to the use of the toolbox TRUETIME. The network traffic is generated using the well-known SAE benchmark. Using this simulation model, the control performance of the active suspension system is studied as a function of the available network bandwidth. *Copyright©2005 IFAC.*

Keywords: Active suspension control, Controller Area Network, Control and real-time scheduling co-design

1. INTRODUCTION

When designing the control laws, the control engineer usually assumes equidistant sampling and actuation (Åström and Wittenmark, 1997). In practise, these assumptions are not met by the implementations of the control laws. In fact, the computations are generally mapped into tasks, which are executed by the real-time operating systems. These tasks are subject to hard real-time constraints ensuring that they finish their execution before a specified deadline. However, ensuring that a task finishes before its deadline does not guarantee that sampling and actuation actions are performed at precise time instants (Martí, 2002). From a control point of view, the implementation introduces varying input output

delays and jitter. Furthermore, when the sensors, the actuators and the controllers of the embedded application are located in distant nodes and interconnected using a communication network, delays and jitter may become more important. Delays and jitter usually degrade the performance of the controlled system and may in extreme cases cause its instability ((Cervin *et al.*, 2003) and (Ben Gaid, Çela and Kocik, 2004)). A simple way of coping with these problems is to choose conservative sampling frequencies. In general, control performance degradations due to delays and jitter are less important when the sampling frequency is increased (see (Seto *et al.*, 1996) and the case study in (Sanfridson, 2004)). However, in the context of embedded systems, and due to economic

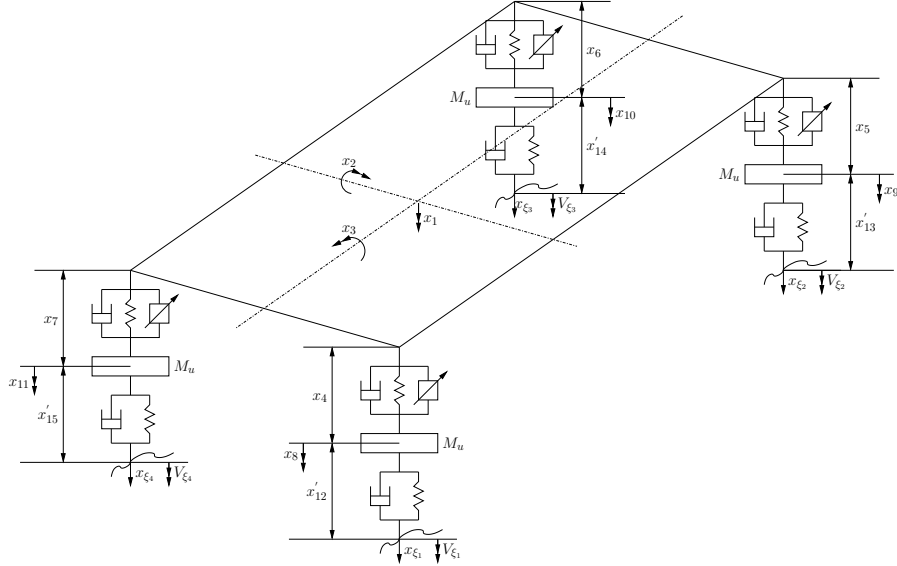


Fig. 1. The suspension system model

and embeddability considerations (i.e. low production cost and small footprint requirements), increasing the sampling frequency may not be possible because the computing and communication resources are subject to resource limitations. When a control application needs to be implemented on a limited resource platform, new design approaches are required. A promising approach to address this problem is the co-design of the control and real-time implementation (Årzén and Cervin, 2005).

In this paper, an active suspension controller, based on a 4-wheels 7 degrees of freedom model is considered, following the classical design methodology of (Chalasani, 1986). Using the Matlab / Simulink based simulator TRUETIME (Andersson *et al.*, 2005), the control performance of its distributed implementation is studied, as a function of the network bandwidth. In order to estimate the true effects of the delays and jitter resulting from the use of a shared CAN bus, the network load induced by the other applications using the bus is generated according to the SAE benchmark (Tindell and Burns, 1994). The aim of this approach is to choose the most efficient implementation platform for a given control application at early design stages in order to reduce the amount of work which is required at the calibration phase, which is very costly in terms of money and time. Evaluating at early stages the performance of a given implementation may lower the number of reiterations of the development cycle, which might be performed in order to correct the control degradations resulting from the implementation choices. This leads to the reduction of development cost and time.

2. THE SUSPENSION CONTROL SYSTEM

In this work, a four-wheels seven degrees of freedom model (figure 1) that was adopted from (Chalasani, 1986) is considered. In this model, the sprung mass (which models the car body), is free to heave, roll and pitch. Note that roll and pitch angles are assumed to be small in order to obtain a linear model. The suspension system connects the car body to the four wheels (front-left, front-right, rear-left and rear-right unsprung masses), which are free to bounce vertically with respect to the sprung mass. A suspension element consists of a spring, a shock absorber and a hydraulic actuator at each corner. The shock absorbers are modeled as linear viscous dampers, and the tires are modeled as linear springs in parallel to linear dampers. The details of the control design can be found in (Chalasani, 1986).

3. THE DISTRIBUTED IMPLEMENTATION OF THE SUSPENSION SYSTEM

The considered controller is based on the full-vehicle model and is implemented on a central processor. In order to estimate the state of the suspension, suspension deflections and unsprung mass velocities of the four corners are sent to the main controller through the CAN. After receiving these measures and computing the control law, the controller sends the control commands to four hydraulic actuators located on the vehicle's corners through the CAN bus. The suspension nodes are synchronized so that the acquisition is performed at the same instant. The computations and the actuations are event triggered. In the next section, these implementation choices will be relaxed and the impact of these relaxations will be studied.

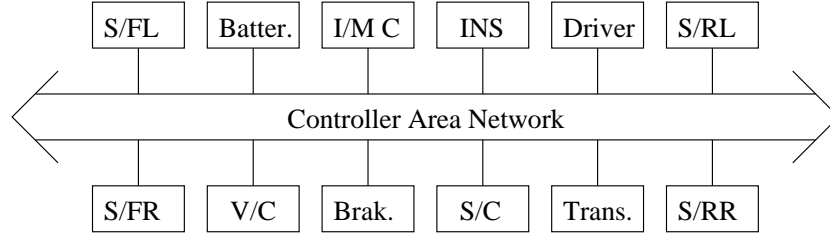


Fig. 2. Architecture

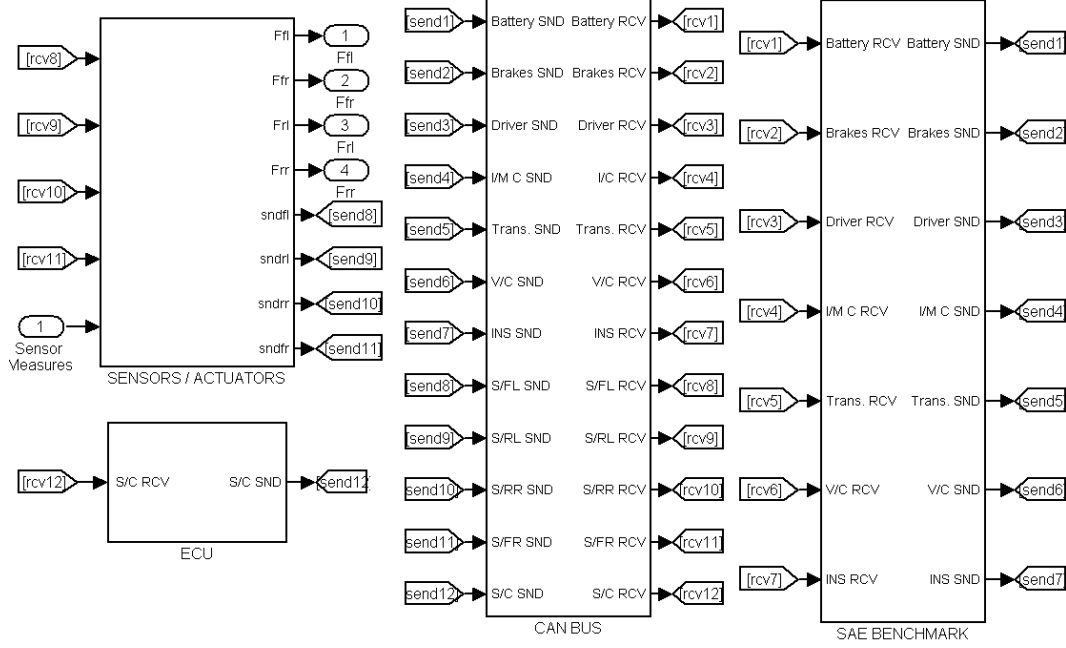


Fig. 3. Implementation Model in TRUETIME. The input port (**Sensor Measures**) and output ports (**Ffl**, **Ffr**, **Frl** and **Frr**) are connected to the suspension model inputs and outputs. The implementation model contains 4 sub-system blocks. The block (**SENSORS/ACTUATORS**) contains the 4 ECUs which perform the sensing and actuation (**S/FL**, **S/FR**, **S/RL** and **S/RR**). The block **ECU** contains the suspension controller (**S/C**). The block (**SAE BENCHMARK**) contains the 7 other ECUs of the car which send the messages over the CAN Bus, which is modelled by the block (**CAN BUS**). Each ECU is simulated using a TRUETIME *Kernel Block*. The CAN bus is simulated using a TRUETIME *Network Block*.

In order to evaluate in an accurate way the effects of the use of a shared CAN bus in the control of the suspension system, a model of the bus traffic have to be used. To this end, the network traffic generated by the other nodes was chosen according to the SAE Benchmark (SAE, 1993). The SAE Benchmark describes the information exchange between seven sub-systems of a prototype electric vehicle. The information which is exchanged is described by a set of 53 *signals*. A signal may be periodic (P) or sporadic (S) and consequently characterized by a period or a minimum inter-arrival time (both designed by T), a relative deadline (D), the source sub-system and the destination sub-system. The seven subsystems are the battery, the vehicle controller (V/C), the inverter/motor controller (I/M C), the instrument panel display (INS), the driver inputs (Driver), the brakes and the transmission (Trans.) control. The simulated

architecture is given in figure 2. An overview of the simulated implementation model in TRUETIME is given in figure 3. The implementation model was described hierarchically. It contains 12 TRUETIME *Kernel Blocks* modelling the involved ECUs and 1 TRUETIME *Network Block* modelling the CAN bus. The modelling of the sub-system block (**SENSORS/ACTUATORS**), which contains the 4 ECUs performing the sensing and actuation (**S/FL**, **S/FR**, **S/RL** and **S/RR**) is described in figure 4

The 53 signals given the SAE benchmark can be found in (Tindell and Burns, 1994). The signals which are exchanged between the distributed components of the suspension control system are listed in table 1.

There are different ways of mapping of the specified signals to CAN messages. In this paper, the

Table 1. Suspension control signals (signals 54 to 62) defined for the particular application developed in this paper

No:	Signals description	Size(bits)	T(ms)	Periodic/Sporadic	D(ms)	From	To
54	Front-Left Suspension Deflection	8	12	P	12	S/FL	S/C
55	Front-Right Suspension Deflection	8	12	P	12	S/FR	S/C
56	Rear-Left Suspension Deflection	8	12	P	12	S/RL	S/C
57	Rear-Right Suspension Deflection	8	12	P	12	S/RR	S/C
58	Front-Left Unsprung mass velocity	8	12	P	12	S/FL	S/C
59	Front-Right Unsprung mass velocity	8	12	P	12	S/FR	S/C
60	Rear-Left Unsprung mass velocity	8	12	P	12	S/RL	S/C
61	Rear-Right Unsprung mass velocity	8	12	P	12	S/RR	S/C
62	Front-Left Control Force	8	12	P	12	S/C	S/FL
63	Front-Right Control Force	8	12	P	12	S/C	S/FR
64	Rear-Left Control Force	8	12	P	12	S/C	S/RL
65	Rear-Right Control Force	8	12	P	12	S/C	S/RR

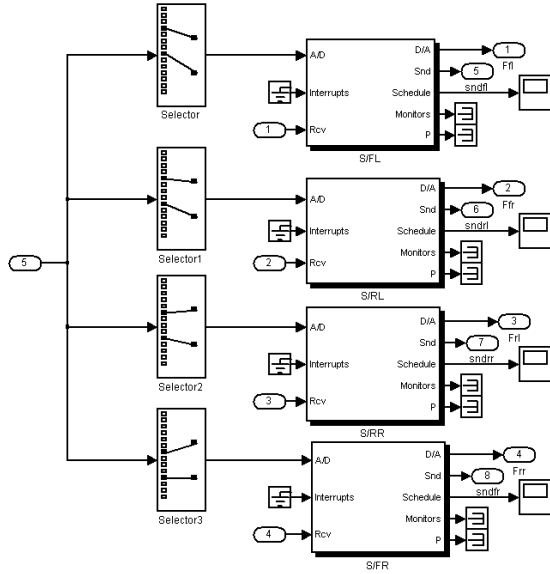


Fig. 4. The sub-system block (**SENSORS/ACTUATORS**), contains the 4 TRUETime *Kernel* blocks, which perform the sensing and actuation and model the ECUs **S/FL**, **S/FR**, **S/RL** and **S/RR**.

work of (Tindell and Burns, 1994) was adopted. In this approach, periodic signals are grouped into the same message if they have the same period and deadline. Server mechanisms are used to transform sporadic messages to periodic ones and to ensure the respect of the deadlines. Using the same approach to the suspension system signals, deflection and velocity measures are grouped into the same message. The control commands to the four actuators can also be grouped into the same message and broadcasted. Table 2 gives the mapping of signals to messages according to this strategy

4. SIMULATION RESULTS

In order to evaluate the performance of the suspension and its implementation, the left side of

Table 2. The mapping of signals to messages in the modified SAE Benchmark. A star (*) indicates a signal that was defined by the SAE Benchmark

Message	Signal	Size (Bytes)	T (ms)	D (ms)
1	*	1	50	5
2	*, *	2	5	5
3	*	1	5	5
4	*, *	2	5	5
5	*	1	5	5
6	*, *, *, *	4	5	5
7	*, *, *, *, *, *	4	10	10
8	*, *, *, *	2	10	10
9	*, *, *, *, *, *	2	10	10
10	*, *, *, *, *	2	10	10
11	54,58	2	12	12
12	55,59	2	12	12
13	56,60	2	12	12
14	57,61	2	12	12
15	62,63,64,65	4	12	12
16	*	1	50	20
17	*, *, *	4	100	100
18	*	1	100	100
19	*	1	100	100
20	*, *, *	3	1000	1000
21	*	1	1000	1000
22	*, *	1	1000	1000

the car is subjected to a “chuck hole” road disturbance (Chalasani, 1986) at the speed of 40 km/h, illustrated in figure 5.

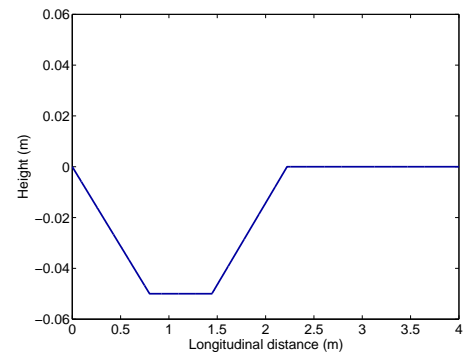


Fig. 5. Chuck hole road disturbance

Simulation results are depicted in figures 6, 7 and 8. They indicate a significant improvement in the control performance of the active suspension with respect to the passive suspension. The control performance resulting from the use of the distributed architecture presents a slight degradation with respect to the ideal simulation. In order to understand the reasons behind this degradation, the network schedule is considered. Observing suspension measures messages (nodes 8 to 11), it can be seen that they present variable time delays, which are due to the use of the network by nodes with higher priority. Note that in these simulations, and in order to respect the hard deadlines of all the messages, network bandwidth was fixed to 250kbps. With this bandwidth, the total utilization of the network is 47 %.

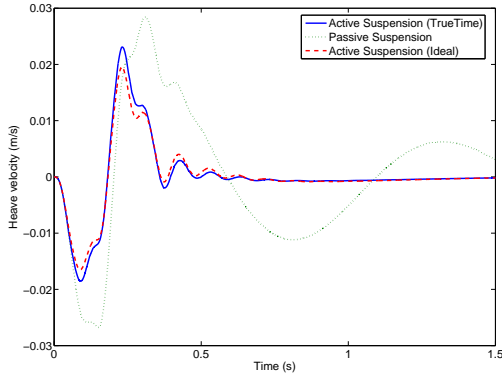


Fig. 6. Heave velocity at 250 kbps with suspension nodes synchronization

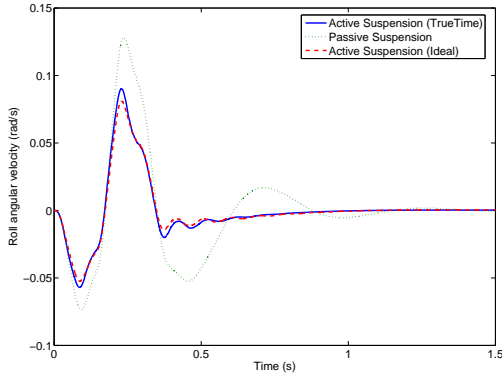


Fig. 7. Roll angular velocity at 250 kbps with suspension nodes synchronization

In order to evaluate the effects of the network bandwidth and suspension nodes synchronization on the control performance, table 3 is constructed. The control performance is measured by the linear quadratic cost functional (which was used to derive the control law according to the LQR method). First, the impact of the network bandwidth on control performance is studied, assuming

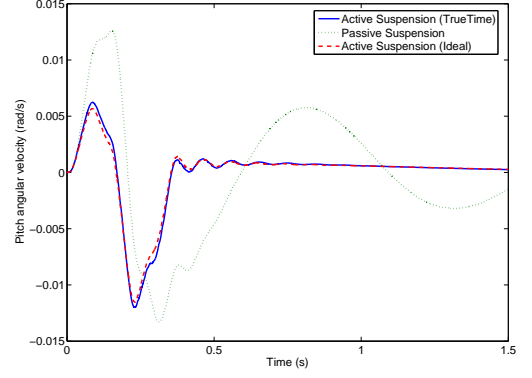


Fig. 8. Pitch angular velocity at 250 kbps with suspension nodes synchronization

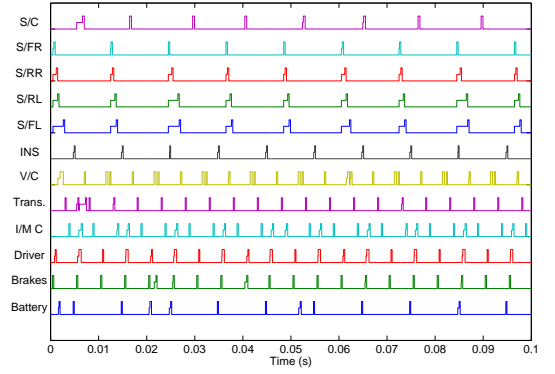


Fig. 9. Network schedule. A high signal means sending, a medium signal means waiting and a low signal means idle

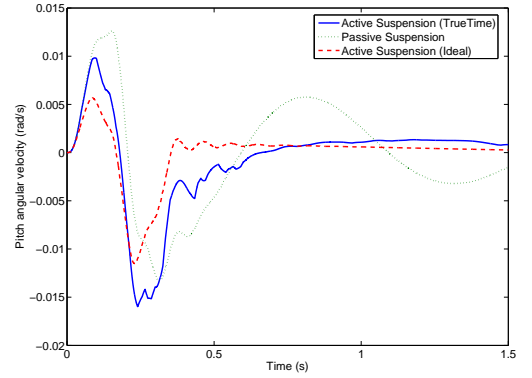


Fig. 10. Pitch angular velocity at 125 kbps with suspension nodes synchronization

suspension nodes synchronization. At 125 kbps, the network utilization is equal to 95 %. Suspension messages may miss their deadlines occasionally. The control performance is degraded (see for example the pitch velocity response in figure 10). From 250 kbps to 1 mbps, all the messages are scheduled before their deadlines. The control performance shows very slight improvements when the network bandwidth increases. Finally, in or-

der to study the impact of the suspension nodes synchronization on the control performance, the preceding study is re-performed assuming that suspension nodes are not synchronized (sampling of the plant is performed periodically but with different offsets between the 4 sensor nodes). The results indicate that for a given network bandwidth, suspension performance is worst when the node are not synchronized

Table 3. Control performance versus network bandwidth

Bandwidth (kbps)	Cost	Network utilization (%)	Sync.
125	181	95	yes
250	139	47	yes
500	134	23	yes
1000	133	12	yes
125	193	95	no
250	149	47	no
500	146	23	no
1000	146	12	no

5. CONCLUSIONS

In this paper, a model of the distributed implementation of a car suspension system was described and simulated. For this particular application, the simulation results performed with TRUE-TIME show that the minimum network bandwidth, which guarantees the schedulability of all the messages, is sufficient to guarantee a good performance of the application. However, the utilization rate of the CAN bus using this approach is 47%. If higher utilization rates are desired, the traditional approaches based on worst-case response time analysis may not be suitable. New scheduling algorithms, which relax the traditional hard-real time constraints (which are traditionally assigned to control applications) and at the same time guarantee a good control performance, are possible candidates, and constitutes a very interesting and promising research direction ((Ben Gaid, Çela and Hamam, 2005) and (Ben Gaid, Çela and Hamam, 2006)).

REFERENCES

Andersson, M., D. Henriksson and A. Cervin (2005). *TrueTime 1.3—Reference Manual*. Department of Automatic Control, Lund University, Sweden.

Årzén, K-E. and A. Cervin (2005). Control and embedded computing: Survey of research directions. In: *Proceeding of the 16th IFAC World Congress on Automatic Control*. Prague, Czech Republic.

Åström, K. J. and B. Wittenmark (1997). *Computer-Controlled Systems : Theory and Design*. Prentice Hall.

Ben Gaid, M-M., A. Çela and R. Kocik (2004). Distributed control of a car suspension system. In: *Proceedings of the 5th EUROSIM Congress on Modeling and Simulation*. Paris, France.

Ben Gaid, M-M., A. Çela and Y. Hamam (2005). Optimal integrated control and scheduling of systems with communication constraints. In: *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*. Seville, Spain.

Ben Gaid, M-M., A. Çela and Y. Hamam (2006). Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system. Accepted for publication in *IEEE Transactions on Control Systems Technology*.

Cervin, A., D. Henriksson, B. Lincoln, J. Eker and K-E. Årzén (2003). How does control timing affect performance?. *IEEE Control Systems Magazine* **23**(3), 16–30.

Chalasani, R. M. (1986). Ride performance potential of active suspension systems Part II: Comprehensive analysis based on a full-car model. In: *Proceedings of the Symposium on Simulation and Control of Ground Vehicles and Transportation Systems, ASME AMD*. Anaheim, California, USA.

Martí, P. (2002). Analysis and Design of Real-Time Control Systems with Varying Control Timing Constraints. PhD thesis. Technical University of Catalonia.

SAE (1993). Class C application requirement considerations. Technical Report J2056/1. Society of Automotive Engineers.

Sanfridson, M. (2004). Quality of control and real-time scheduling: Allowing for time-variations in computer control systems. PhD thesis. Department of Machine Design, Royal Institute of Technology.

Seto, D., J. P. Lehoczy, L. Sha and K. G. Shin (1996). On task schedulability in real-time control systems. In: *Proceedings of the 17th IEEE Real-Time Systems Symposium*. New York, USA.

Tindell, K. and A. Burns (1994). Guaranteeing message latencies on control area network (CAN). In: *Proceedings of the 1st International CAN Conference*. Mainz, Germany.